Electronics and Electrical Engineering

https://www.aimspress.com/journal/ElectrEng

AIMS Electronics and Electrical Engineering, 9(4): 541–564.
DOI: 10.3934/electreng.2025024
Received: 01 May 2025
Revised: 03 August 2025
Accepted: 01 September 2025
Published: 05 September 2025

Research article

AI-driven equipment scheduling under variable electricity pricing: A case study on dryer cluster efficiency and standby capacity planning

Guoliang Feng[1], Tianren Gao[1], Shaojun Bian[2] and Tianming Yu[1,*]

[1] School of Automation Engineering, Northeast Electric Power University, Jilin City, People's Republic of China

[2] School of Creative and Digital Industries, Buckinghamshire New University, High Wycombe HP11 2JZ, UK

* **Correspondence:** Email: 20192928@neepu.edu.cn.

Abstract: As contemporary industrial enterprises and manufacturing facilities expand in scale, the judicious scheduling of numerous in-house equipment clusters has become a critical factor in enhancing operational efficiency and reducing production costs. Traditional static algorithms merely sustain the baseline functionality of factory apparatuses, thereby failing to unlock the latent synergy among integrated equipment. In response to the advent of artificial intelligence and the maturation of machine learning algorithms, innovative solutions have emerged to address such operational scheduling challenges. This paper presents an algorithm predicated on a deep reinforcement learning-based optimization strategy, integrated with the localized time-of-use electricity tariff framework, to diminish the operating expenses of dryer clusters within the facility. Initially, an objective function aimed at minimizing total operating costs was constructed, alongside the formulation of constraint conditions derived from the operational protocols of the dryer clusters. Subsequently, under a reinforcement learning paradigm, the dryer clusters were modeled by delineating the state and action spaces and formulating a reward function contingent upon the tariff schedule and the degree of completion of gas-drying tasks. Ultimately, a neural network employing a double-DQN architecture was developed and utilized to resolve the scheduling strategy through model training. Empirical results from multiple training iterations with dryer clusters validated that the proposed algorithm can reduce electricity expenditures without necessitating alterations to the existing equipment, while also providing recommendations for the optimal number of additional standby dryers to further reduce operational costs.

Keywords: multi-agent systems; industrial operations scheduling; deep reinforcement learning (DRL)

1. Introduction

As modern manufacturing facilities evolve, production workflows have grown increasingly intricate, accompanied by a proportional rise in the number of required production machines. The judicious resolution of flexible job shop scheduling problems (FJSP) [1] within such environments directly impacts both production efficiency and operating costs. In essence, FJSP involve the optimal allocation of tasks to constrained resources—such as machinery and labor—with the objective of minimizing key performance metrics, including total processing time, delays, and overall expenses.

Regarding the classical optimization problem of factory scheduling, numerous scholars have employed a variety of methods to address it. These methods encompass many traditional optimization algorithms, including genetic algorithms [2], particle swarm optimization [3], ant colony optimization [4], and their hybrid variants [5, 6]. In reference [7], a mixed-integer programming model was proposed for the flexible job shop scheduling problem with job splitting, wherein a complete job is partitioned into several batches to minimize the overall job completion time. A genetic algorithm augmented with local search was utilized to solve the model, and its feasibility was subsequently demonstrated in comparison with a conventional genetic algorithm. In reference [8], in addition to accounting for the intrinsic processing times on machines, the authors also considered inter-machine transfer times and machine setup times. An improved genetic algorithm was developed to minimize processing, setup, and transfer times by employing three distinct methods to generate initial solutions and utilizing a manually paired crossover strategy for solution refinement. Experimental results validated the superiority of their approach. In reference [9], a variable neighborhood-enhanced genetic algorithm was introduced for the job shop scheduling problem, wherein elements from particle swarm optimization were integrated into the genetic algorithm framework to mitigate the slow convergence and suboptimal precision associated with traditional genetic algorithms. Moreover, encoding, decoding, and crossover operations were implemented using real numbers. Comparative experiments affirmed the effectiveness of their method. In reference [10], an integrated approach was proposed for a job shop scheduling problem that also considers the scheduling of transportation vehicles for moving workpieces among machines. An objective function was formulated to minimize both the makespan and earliness, and a particle swarm optimization algorithm combined with simulated annealing was employed alongside a rapid lower bound procedure to address this hybrid problem. Extensive experimental validations revealed that this algorithm consistently achieved cost reductions and ensured timely deliveries. In reference [11], a highly complex factory environment was modeled using a mixed-integer linear programming formulation that encompassed multiple parallel, unrelated machines, various flexibly scheduled tasks, and several specialized workpiece transporters. The research team integrated parallel genetic algorithms and parallel particle swarm optimization, and two statistical tests substantiated the reliability and efficacy of their approach. In reference [12], factory scheduling was combined with the path planning of workpiece transport vehicles, with an objective function aimed at minimizing production intervals and transportation durations. An ant colony optimization algorithm enhanced with local search was employed for scheduling optimization, while a separate programming solution addressed the path planning component. Experimental validation confirmed that this integrated approach significantly improved factory operational efficiency. In reference [13], a distributed job shop scheduling problem was addressed by allocating a complete job across multiple factories and devising operational schedules for each. With the objective of

minimizing the job intervals across all facilities, an ant colony optimization algorithm was applied, ultimately yielding a commendable scheduling solution. In addition to the literature described above, numerous other studies have attempted to solve factory scheduling problems using traditional optimization algorithms such as genetic algorithms [14, 15], particle swarm optimization [16, 17], and ant colony optimization [18, 19]. In summary, although these traditional optimization algorithms have achieved notable results in their respective applications, inherent limitations—such as entrapment in local optima, slow convergence, reduced precision, and difficulties in adapting to higher-dimensional models—remain. Consequently, addressing these challenges necessitates the introduction of novel algorithms that integrate two or more existing methods, which inevitably increases algorithmic complexity and complicates parameter tuning.

With the burgeoning development of artificial intelligence, machine learning [20] has been increasingly embraced by researchers across diverse disciplines. Similarly, machine learning has found extensive application in factory scheduling, offering significant advantages in its resolution. At its core, machine learning is an algorithmic methodology that involves training models on data to enable the prediction, classification, and processing of new information. Traditional machine learning approaches can be broadly categorized into three primary types: supervised learning [21], unsupervised learning [22], and semi-supervised learning [23]. Reinforcement learning [24], a distinct branch of machine learning, diverges from these paradigms by eschewing the development of static input-to-output mappings; instead, it focuses on acquiring optimal policies through trial-and-error interactions with the environment and subsequent feedback. For example, while supervised learning relies on annotated datasets for training, reinforcement learning depends on reward signals. In this framework, an agent learns to execute actions in specific contexts by interacting with the environment to maximize cumulative rewards, continually refining its strategy through iterative trial and error to identify the optimal decision-making pathway. By applying reinforcement learning to factory scheduling, the complex challenge of devising intricate scheduling strategies can be reformulated into the task of constructing an intelligent agent and simulating the real environment, thereby facilitating the derivation of an effective scheduling policy. Overall, this approach has the potential to reduce computational complexity and enhance algorithmic stability in the face of unforeseen events. Deep learning [25] represents a specialized subset of machine learning. Although conventional machine learning algorithms—such as linear regression and support vector machines—are capable of extracting patterns from data through various techniques, deep learning distinguishes itself by employing neural networks to process complex data patterns and structures. Leveraging the potent representational capacity of multi-layered neural networks, deep learning has demonstrated exceptional performance in domains such as computer vision [26], speech processing [27, 28], industrial manufacturing [29, 30], and scientific research [31, 32]. When applied to factory scheduling problems, deep learning can accelerate convergence, enhance operational efficiency, and reduce the time required to obtain scheduling solutions. Consequently, the integration of deep reinforcement learning [33], which amalgamates the strengths of both deep learning and reinforcement learning, has emerged as a promising approach to mitigate the limitations of traditional algorithms and more effectively address factory scheduling challenges. Recent advancements underscore the growing efficacy of deep reinforcement learning (DRL) in complex scheduling domains. As comprehensively surveyed by Lv et al. [34], DRL is rapidly becoming a key methodology for job shop scheduling problems. This trend is further evidenced by practical applications, such as the work of Wang et al. [35], who successfully employed a multi-agent

DRL framework to achieve dynamic, energy-efficient scheduling of integrated production and logistics systems. The core principle of using multi-agent systems to optimize scheduling under dynamic pricing has also been validated in adjacent fields, as demonstrated by Zulfiqar et al. [36] in the context of electric vehicle charging. Our research builds upon this solid foundation, applying a tailored multi-agent DRL approach to enhance the operational efficiency of industrial equipment clusters.

The operational scheduling of such equipment clusters presents a unique set of challenges that traditional methods struggle to address effectively. While classical optimization algorithms, such as genetic algorithms or particle swarm optimization, can find high-quality static schedules, they often lack the flexibility to adapt to the dynamic, stochastic nature of real-world factory operations and can be computationally expensive for high-dimensional problems. On the other hand, a standard single-agent reinforcement learning approach is not a natural fit, as it would require a centralized controller with a prohibitively large state-action space, encompassing the status of every single unit. The problem is more accurately characterized as a cooperative multi-agent system (MAS), where each piece of equipment (or dryer) can be modeled as an autonomous agent. Each agent must make its own decisions (e.g., when to enter regeneration) while coordinating with others to achieve a collective goal—namely, ensuring the facility's production demand is always met. This inherent structure strongly motivates the adoption of a multi-agent deep reinforcement learning (MA-DRL) framework. MA-DRL is particularly well-suited to this domain because it allows for decentralized execution, handles complex interactions between agents, and can learn sophisticated, cooperative policies through trial-and-error interaction with a simulated environment, making it a powerful and scalable solution for our problem.

This paper presents a deep reinforcement learning-based factory scheduling algorithm aimed at optimizing the scheduling of dryer clusters within a facility. A mathematical model is formulated with an objective function that minimizes total operational costs, subject to constraints imposed by the local electricity tariff schedule, the operational protocols of the dryer clusters, and the gas drying task requirements of the facility. Within the deep reinforcement learning framework, the state space is defined by the internal conditions of the dryer clusters, the action space is delineated based on the feasible operations of these clusters, and a reward function is established in accordance with the local tariff schedule and the completion status of the gas drying tasks. A deep Q-network is then employed to train the model and derive the scheduling optimization strategy. Furthermore, to further reduce overall operational costs, the algorithm also provides recommendations regarding the optimal number of additional standby dryers by iteratively applying the aforementioned methodology to balance the cost of procuring extra dryers against the savings in electricity expenses, considering both the total gas drying operational time and the cost per dryer. Ultimately, simulation experiments have validated the efficacy and reliability of the proposed algorithm.

The main contributions and novelties of this paper are summarized as follows:

- **Novel Problem Formulation:** We are the first to formulate the industrial dryer cluster scheduling problem under variable electricity pricing as a cooperative multi-agent reinforcement learning task. This paradigm shift from a centralized to a decentralized optimization approach better reflects the operational nature of the equipment.
- **Hybrid Reward Structure Design:** We propose a novel, two-tiered reward structure that effectively balances competing objectives. It combines an individual reward for each agent, encouraging local cost minimization based on TOU tariffs, with a global reward for the entire system, enforcing the critical constraint of meeting collective production demands.

- **Integrated Scheduling and Capacity Planning:** Our framework provides a dual-purpose solution. Beyond generating an economically optimized daily operational schedule, it also serves as a strategic planning tool. By simulating scenarios with varying numbers of standby units, the algorithm provides data-driven recommendations for optimal long-term standby capacity, bridging the gap between operational scheduling and capital investment decisions.

We validate the proposed algorithm's effectiveness and resilience through extensive simulations, including scenarios with dynamic changes in production demand and unexpected equipment failures, demonstrating its practical applicability in real-world industrial settings.

2. Mathematical model of the drying unit ensemble

The objective of optimizing dryer cluster scheduling is to substantially reduce the operational costs associated with gas drying processes. Specifically, without compromising the facility's standard gas drying tasks or breaching the operational constraints of the dryers, an optimal scheduling plan is devised solely by managing the start-up and shut-down sequences of multiple dryers within the cluster to minimize total electricity expenses. Furthermore, the algorithm provides recommendations for the integration of additional standby dryers and formulates corresponding scheduling plans, based on the existing scale of the dryer cluster, to further diminish overall operating costs.

2.1. Objective function

We construct a mathematical model for the dryer cluster with an objective function aimed at minimizing total operational costs. The total cost comprises the electricity expenses for the gas drying process, the electricity expenses for the tank heating regeneration phase, and the unit cost of acquiring additional dryers. In formulating the objective function, we have deliberately focused on the variable electricity consumption costs associated with the TOU (time-of-use) tariff and the procurement cost of additional equipment. It is important to acknowledge another significant component of industrial electricity billing: the grid capacity cost, which is determined by the facility's peak power demand within a billing cycle. While managing peak demand is critical in many industrial optimization problems, its impact is considered negligible for the specific case of the dryer cluster studied herein. The operational power of each dryer during its primary gas-drying task is relatively low (under 10 kW), and the power for the more energy-intensive regeneration phase is controlled at a consistent level of approximately 200 kW. Given that a large industrial facility's peak demand is typically dictated by much larger, intermittently operating machinery, the scheduling of a single dryer's regeneration is unlikely to establish a new facility-wide peak. Consequently, the primary lever for cost reduction in this context is the strategic scheduling of regeneration cycles to coincide with low-price electricity periods. Therefore, our model prioritizes the optimization of energy consumption costs, which presents the most substantial opportunity for savings. Given that the facility's operational requirements mandate the continuous availability of a requisite number of dryers for the gas drying process at all times, the electricity cost for this process remains fixed. Consequently, the adjustable operational costs are limited to the electricity cost incurred during the tank heating regeneration phase and the unit cost associated

with additional standby dryers, as detailed below:

$$W_D = \sum_{t=1}^{T} \sum_{i=1}^{I} E_{i,t} + \sum_{k=1}^{K} W_k \tag{2.1}$$

where W_D represents the total adjustable cost, T denotes the overall duration of gas drying operations, I signifies the total number of drying units, K indicates the total number of additional standby drying units, $E_{i,j}$ is the electricity cost incurred by the regeneration process of the i-th drying unit at time t, as presented in Equation (2.2), and W_K corresponds to the cost associated with the k-th additional standby drying unit.

$$E_{i,t} = S_{1,i,t} \times T_{1,i,t} + S_{2,i,t} \times T_{2,i,t} \tag{2.2}$$

where $S_{1,i,t}$ and $S_{2,i,t}$ denote the state values of tanks 1 and 2 within the i-th drying unit at time t, each adopting binary states—0 indicating deactivation and 1 indicating activation, $T_{1,i,t}$ and $T_{2,i,t}$ correspond to the electricity costs incurred by the tank heating regeneration process for tanks 1 and 2 in the i-th drying unit at time t.

2.2. *Constraints*

The principal constraints for the optimization of dryer cluster scheduling include the operational protocols of the dryers, the maximum allowable duration for gas drying operations within the tanks, the minimum required duration for tank heating regeneration, the facility's task requirements, and the local electricity tariff schedule.

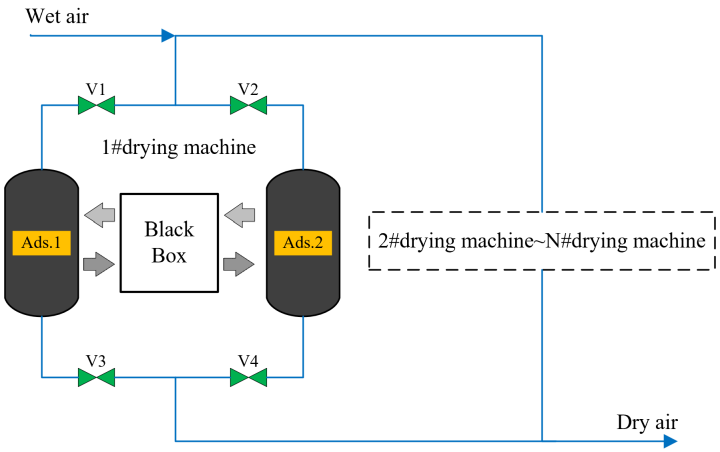

Figure 1. Simplified diagram of the drying unit ensemble system structure.

Figure 1 presents a simplified schematic of the internal system architecture of the dryer cluster, which comprises several dryers arranged in parallel. Each dryer contains two tanks, designated as $Ads.2$ and $Ads.1$. Both tanks are capable of executing gas drying operations and heating regeneration processes; however, they cannot perform both operations concurrently. The black box includes components such as fans, heaters, and coolers, which have been omitted due to their irrelevance to this study.

Maximum gas drying operation duration for the tank:

$$T_D < T_1 \tag{2.3}$$

In the equation, T_D denotes the actual gas drying operation duration of the two tanks within the dryer, while T_1 represents the maximum allowable gas drying duration for a tank. Due to the limitations imposed by the adsorptive materials within the tanks, only the dried gas produced within the prescribed maximum duration satisfies production requirements. Once this duration is exceeded, the internal materials become deactivated, necessitating that the tank be either shut down or placed in standby mode. The tank may only resume gas drying operations after undergoing a heating regeneration process, which reactivates the internal materials.

Minimum tank heating regeneration duration:

$$T_R < T_2 \tag{2.4}$$

In the equation, T_R represents the actual duration of the tank heating regeneration process for the two tanks within the dryer, while T_2 denotes the minimum permissible duration for this process. Due to limitations of the adsorptive materials within the tanks, once deactivated, these materials cannot continue to support the gas drying operation. Only after undergoing the prescribed period of tank heating regeneration do the materials regain their activity, thereby enabling the tanks to be restarted and reengaged in gas drying operations.

Factory task requirements:

$$N = \sum_{i=1}^{I} (S_{1,i,t} + S_{2,i,t})\ t \in \bigcup \tag{2.5}$$

In the equation, N denotes the number of dryers required under the operating conditions, while U represents the universal set encompassing any time instance within the total operating period. At any given moment, a specified number of dryers (i.e., an equivalent number of tanks) are actively engaged in gas drying operations.

Electricity tariff partition:

$$T_{1,i,t} \ and \ T_{2,i,t} = \begin{cases} c_1\ , t \in \omega_1 \\ c_2\ , t \in \omega_2 \\ c_3\ , t \in \omega 3 \end{cases} \tag{2.6}$$

where ω_1, ω_2 and ω_3 represent the peak, flat, and valley intervals, respectively, from the local electricity tariff partition table; c_1, c_2 and c_3 correspond to the electricity costs associated with each of these periods.

3. Problem description under the reinforcement learning framework

This paper employs a deep reinforcement learning approach to derive an optimized scheduling strategy for dryer clusters. Traditional reinforcement learning is primarily composed of three key elements: the state space, the action space, and the reward space. In this study, the state space is defined by the statuses of multiple tanks within the dryer cluster along with their corresponding time

nodes. The intelligent agent selects actions based on its current policy and the state space, while the reward space yields feedback corresponding to the changes in the state space induced by these actions. The agent iteratively refines its policy based on the accumulated data, ultimately converging on a scheduling strategy that effectively reduces electricity costs.

3.1. State space

For the dryer cluster scheduling optimization problem addressed in this paper, the state space is encapsulated by a state table. Table 1 illustrates the state table for a single dryer, wherein the row labels "adsorber1" and "adsorber2" correspond to the two tanks within the dryer, and the column labels denote discrete time nodes. The table entries are categorized into three types—"0", "work", and "regen"—which represent, respectively, a waiting state, active gas drying, and the heating regeneration process.

Table 1. Drying unit state table.

state	adsorber1	adsorber2
0	work	0
1	work	0
2	work	0
3	work	0
4	work	0
5	work	regen
6	work	regen
7	work	regen

3.2. Action space

For the dryer cluster scheduling optimization problem, this paper adopts a deep reinforcement learning approach. Specifically, the action space A for a single dryer agent comprises three actions: regen1, regen2, and delay, corresponding to the heating regeneration operation of tank 1, the heating regeneration operation of tank 2, and no action, respectively. For a dryer cluster containing N dryers, the agent's output consists of $3N$ action combinations, which can be expressed as:

$$A = \{A_G\} \tag{3.1}$$

where A_G denotes the set of action output combinations for N drying units.

3.3. Reward function

The total reward value r_i for the intelligent agent comprises two components: individual rewards and overall rewards, as follows:

$$r_i = r_{s,t} + r_{g,d} \tag{3.2}$$

where $r_{s,t}$ denotes the individual reward value for the s-th drying unit at time t, as presented in Equation (3.3); $r_{g,d}$ represents the overall reward value corresponding to d instances of insufficient drying units

upon the completion of a cycle, as specified in Equation (3.4).

$$r_{s,t} = \begin{cases} r_1 \ , t \in \omega_1 \\ r_2 \ , t \in \omega_2 \\ r_3 \ , t \in \omega 3 \end{cases} \tag{3.3}$$

where r_1, r_2 and r_3 represent the reward values corresponding to the peak, flat, and valley periods, respectively. Within each cycle, the intelligent agent outputs individual rewards at every time instance based on the electricity tariff partition table, with these reward values being inversely correlated to the factory's local electricity tariff segmentation.

$$r_{g,d} = \begin{cases} r_4, \ d = 0 \\ r_5, \ 0 < d \leq d_1 \\ r_6, \ d_1 < d \leq d_2 \\ \vdots \\ r_n, \ d_{n-5} < d \leq d_{n-4} \end{cases} \tag{3.4}$$

where r_4 through r_n denote the stepwise overall reward values, and d_1 through d_{n-4} represent the defined partition points for insufficient drying unit occurrences. Prior to the completion of a full cycle, the intelligent agent issues an overall reward, the value of which is correlated with both the successful execution of the factory's gas drying operations and the cooperative performance among the drying units.

4. Method for solving the drying unit ensemble scheduling strategy

In conventional reinforcement learning, Q-learning utilizes a Q-table to store state-action pairs, denoted as $Q(s, a)$. The update equation for the Q-value is presented as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \tag{4.1}$$

where s represents the current state, a denotes the current action, r indicates the immediate reward, s' signifies the subsequent state, γ is the discount factor, and α represents the learning rate. In the equation, $\max_{a'} Q(s', a')$ denotes the maximal Q-value among all possible actions in state s', which is employed to estimate the future returns of the optimal policy.

However, traditional Q-learning based on a Q-table is more suited to problems characterized by a limited number of environmental states and low dimensionality. When the state dimensionality increases, it can lead to the curse of dimensionality. Therefore, this paper incorporates neural networks by employing a deep Q-network to address the optimization problem of scheduling the drying unit ensemble. By substituting the Q-table with a neural network, the algorithm transitions from Q-table optimization to neural network training.

The essence of neural network training is the minimization of the loss function $L(eval)$, also known as the mean squared error, as defined by the following equation:

$$L(eval) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{4.2}$$

where $\widehat{y_i}$ denotes the predicted value for the i-th data sample, as delineated in Equation (4.3).

$$\widehat{y_i} = Q_{eval}(s_i, a_i) \tag{4.3}$$

where y_i represents the true value of the i-th data sample, as shown in Equation (4.4).

$$y_i = r_1 + \gamma Q_{target}(s_{i+1}, a_{i+1}) \tag{4.4}$$

where a_{i+1} denotes the action corresponding to the maximum reward value obtained by the Q_{eval} network for the subsequent state s_{i+1}, as shown in Equation (4.5).

$$a_{i+1} = argmax_a Q_{eval}(s_{i+1}, a) \tag{4.5}$$

The deep Q-network structure established in this paper is illustrated in Figure 2, which meticulously delineates the parameter update process of the target network. Initially, the evaluation network extracts state s_i and action a_i from the training dataset and outputs the predicted value to the loss function. Subsequently, the evaluation network retrieves the subsequent state s_{i+1} from the training dataset and determines the corresponding action a_{i+1}, which is expected to yield the maximum reward for s_{i+1}, and forwards it to the target network. The target network then acquires s_{i+1} from the training dataset, and by combining the action a_{i+1} from the evaluation network with the reward r_i from the training set, it produces the true reward value to be fed into the loss function. Finally, the loss function performs gradient descent to update the internal parameters of the evaluation network, while the target network's parameters are periodically synchronized with those of the evaluation network.

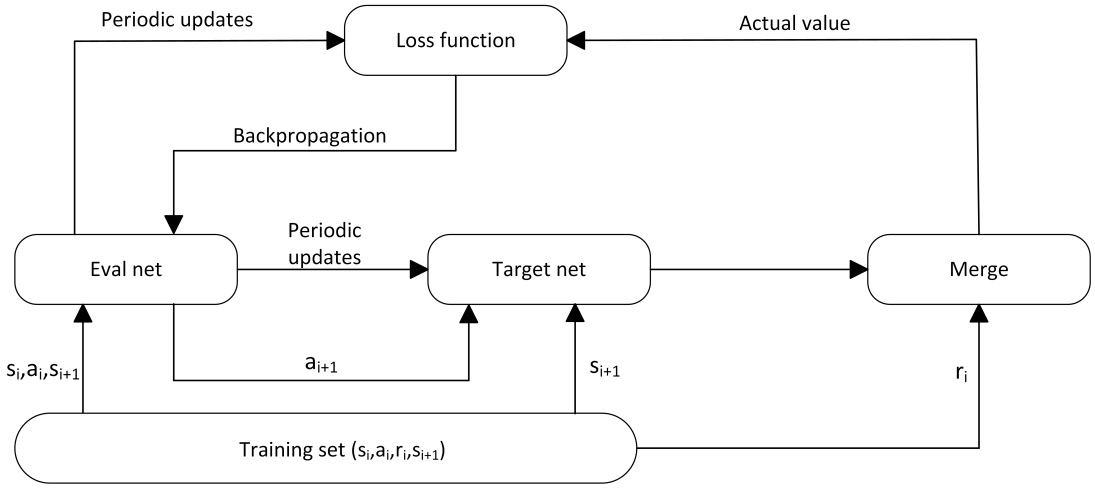

Figure 2. Deep Q-network system architecture.

The overall framework for the DQN-based optimization strategy for scheduling the ensemble of drying units is illustrated in Figure 3. As depicted, the algorithm assigns an identical architecture to each drying unit within the ensemble—each unit is equipped with its own neural network, individual reward function, experience pool, and training dataset. When all drying units are of the same model, the internal parameters of these components are identical; conversely, if the models differ, the corresponding internal parameters will vary. Taking Drying Unit 1 as an example, the scheduling

solution process for the other units is analogous: initially, the double-DQN network outputs an action based on the current policy; the drying unit then transitions its state according to this action and stores the state s_i, action a_i, and subsequent state s_{i+1} in the experience pool. Next, the individual reward function computes a reward value, $r_{s,t}$, based on the state transition and records it in the experience pool. This cycle is repeated until the completion of a full cycle, at which point the overall reward function evaluates the collaborative task performance among the drying units and outputs an overall reward value, $r_{g,d}$, into the experience pool (the reward update method in the experience pool can be referenced in Equation (3.2)). Subsequently, once a sufficient amount of data has accumulated in the experience pool, a training dataset is randomly sampled and fed into the double-DQN for training (the specific training methodology is described in Figure 2). Finally, upon reaching the preset number of cycles and achieving substantial electricity cost savings, the actions output by the neural networks corresponding to all drying units are consolidated to yield the final optimized scheduling strategy for the drying unit ensemble.

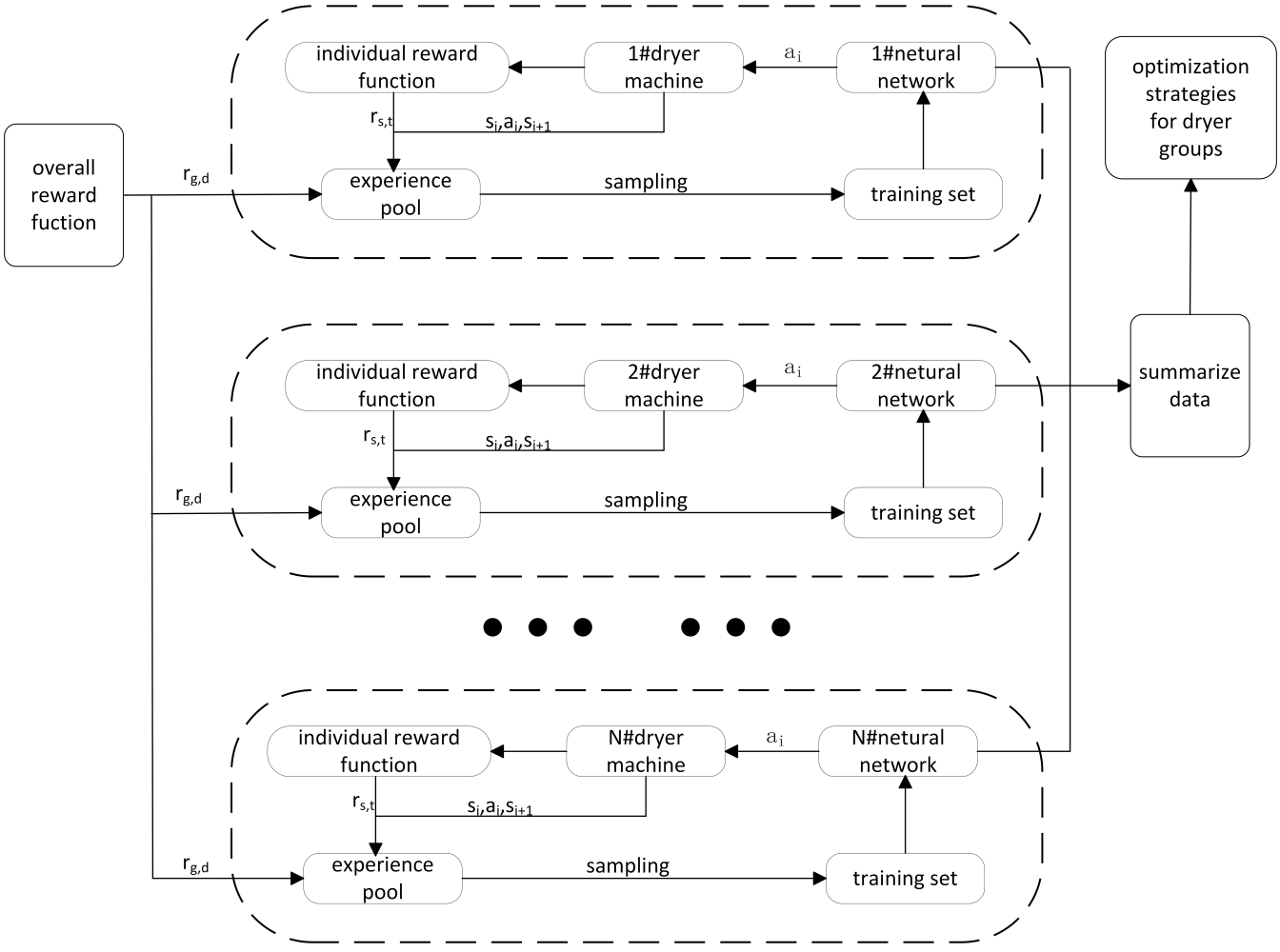

Figure 3. Overall framework for the drying unit ensemble scheduling optimization strategy.

Algorithm 1 depicts the procedure of the drying unit ensemble scheduling optimization strategy.

Algorithm 1 The drying unit ensemble scheduling optimization

Initialize for all agents $i \in \{1, ..., N\}$: evaluation networks $Q_{\text{eval},i}(\cdot; \theta_i)$, target networks $Q_{\text{target},i}(\cdot; \theta'_i)$ with $\theta'_i \leftarrow \theta_i$, and individual replay buffers D_i.
for episode = 1 to M **do**
  Reset environment and receive initial state s_1.
  for timestep t = 1 to T_{cycle} **do**
    For each agent i, select action $a_{i,t}$ using an ε-greedy policy on $Q_{\text{eval},i}(s_t, \cdot)$.
    Execute joint action $a_t = (a_{1,t}, ..., a_{N,t})$ and observe next state s_{t+1}.
    for each agent i = 1 to N **do**
      Calculate individual reward $r_{s,i,t}$ and store $(s_t, a_{i,t}, r_{s,i,t}, s_{t+1})$ in D_i.
    end for
    $s_t \leftarrow s_{t+1}$.
  end for
  Calculate global reward $r_{g,d}$ for the completed episode.
  for each agent i = 1 to N **do**
    Add $r_{g,d}$ to the reward of the final transition stored in D_i.
    Sample a minibatch from D_i.
    Calculate target values y_j for the minibatch using the DDQN rule:
      $y_j \leftarrow r_j + \gamma Q_{\text{target},i}(s'_j, \arg\max_{a'} Q_{\text{eval},i}(s'_j, a'; \theta_i); \theta'_i)$
    Update weights θ_i by performing a gradient descent step on loss $\mathcal{L}(\theta_i) = \text{MSE}(y, Q_{\text{eval},i})$.
  end for
  if episode mod $F_{\text{update}} == 0$ **then**
    $\theta'_i \leftarrow \theta_i$ for all agents i.
  end if
end for

5. Analysis of simulation experiments

In this study, a simulation experiment is conducted on a drying unit ensemble system comprising 7 identical drying units in a factory to validate the algorithm's efficacy. The parameter settings for the ensemble include: five drying units required for operational conditions, a cycle duration of 180 days, a maximum gas drying operation duration T_1 of 8 hours for the tanks, a minimum heating regeneration duration T_2 of 3 hours for the tanks, and the local electricity tariff partition table as detailed in Table 2.

Table 2. Electricity tariff time division chart.

Time Period Name	Time Division	Electricity Price (Yuan/kWh)
Peak Hours	11:00-12:00 14:00-21:00	1.015
Normal Hours	7:00-11:00 12:00-14:00	0.628
Off-Peak Hours	21:00-23:00 23:00-7:00	0.242

The hyperparameter settings for the DQN algorithm are presented in Table 3.

Table 3. Algorithm hyperparameters.

Project	Parameters
Learning Rate	0.01
Reward Discount	0.9
Neural Network Architecture	The hidden layer count is 7, with each layer comprising 50 neurons.
Activation Function	Sigmoid function
Total Number of Training Iterations	300
Experience Replay Pool Capacity	4320

5.1. Training analysis

The average scheduling reward values from the first 100 cycles were recorded and the resulting curve was smoothed, as depicted in Figure 4. It is evident that during the initial exploration phase, the agent achieved a very low average reward with a marked downward trend, attributable to its nascent and suboptimal scheduling strategy. As the number of iterations increased, the average scheduling reward rapidly ascended and converged to a fixed level, driven by the influence of the individual reward function, which steers the agent's decisions toward optimality and minimizes the electricity cost per dryer. Even when the curve stabilized, inherent fluctuations persisted—a normal occurrence. In the later stages of training, the agent continues to explore the action space with a certain probability, which may occasionally result in decisions that yield lower rewards. Moreover, due to the operational constraints of each dryer and the variability in start-stop timings across training iterations, the internal state of a dryer at a given time node may differ, leading to distinct cost-optimal action decisions. These factors collectively contribute to the oscillatory behavior observed in the average total return curve.

Figure 5 depicts the loss function curve. Initially, both the loss values and the oscillation amplitudes are considerable; however, as the number of training iterations increases, the curve progressively converges toward the horizontal axis with a marked reduction in oscillation amplitude. This indicates that the neural network is increasingly well-fitted to the real environment.

Figure 6 illustrates the curve representing the frequency of insufficient dryers. "Insufficient dryers" refers to instances during one or more time periods within a cycle where the number of dryers in operation is less than the number required by the operating conditions. In the initial training phase, the agent is in an exploratory stage, and the internal model parameters are not yet well-tuned; thus, the selection and output of actions are largely random. Furthermore, during the early stages of training, each agent primarily focuses on minimizing the electricity cost of its corresponding dryer, often neglecting the cooperative completion of the gas drying task, which results in significant fluctuations in the first half of the curve. As the experience replay buffer accumulates more sample data, the agents' training updates progressively steer toward achieving higher rewards and better cooperation in fulfilling the gas drying tasks, thereby reducing the frequency of insufficient dryer instances. Once the agents gradually stabilize, the frequency of insufficient dryers consistently converges to zero, with only an extremely low probability of non-zero occurrences due to the maintained exploration probability in the later training stages.

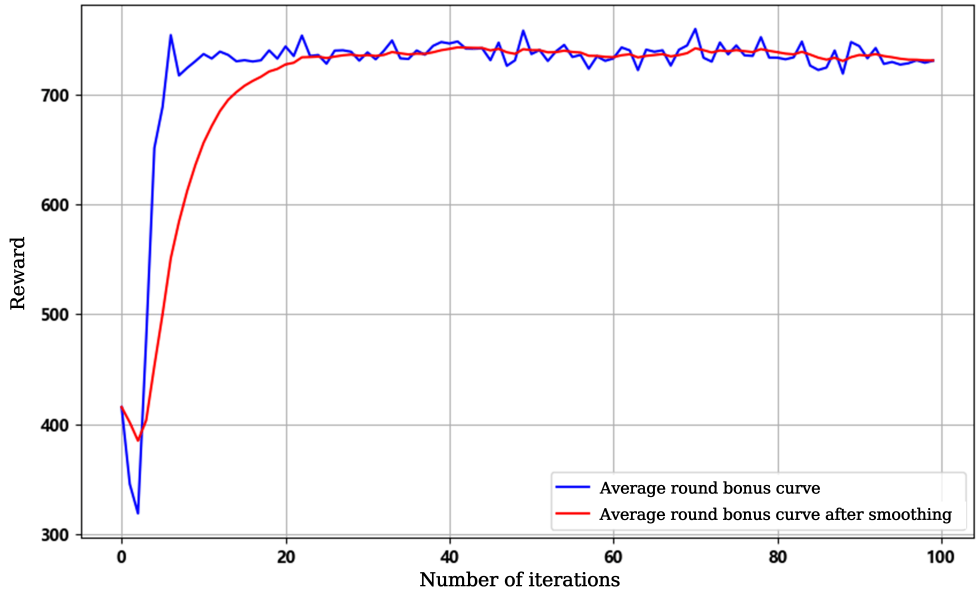

Figure 4. Average scheduling reward value.

5.2. Scheduling result analysis

After 500 training iterations, the aggregated information and scheduling outcomes are presented. Table 4 illustrates the algorithm's recommendations for the number of additional standby dryers along with the resulting cost reductions, based on the factory's current state. As shown in Table 4, under the scenario of 7 existing dryers on-site and an operational requirement for 5 dryers, the algorithm experimented with four different combinations. Among these, the third combination achieved the greatest total cost reduction over a six-month period. Although this recommendation incurs the cost of an additional standby dryer, the electricity cost savings from optimized scheduling are sufficient to offset this expense. When the cycle duration is extended to one year, the third combination still yields the highest overall cost reduction; however, the gap in electricity savings between the third and fourth combinations gradually diminishes. Extending the cycle duration further results in the outcomes depicted in Figure 7.

Table 4. Optimal electricity cost savings chart.

Total Number of Standby Dryers	Number of Additional Standby Dryers	Cost of Additional Standby Dryers	Optimized Electricity Cost (Yuan)	Six-Month Total Cost Reduction (Yuan)	First-Year Total Cost Reduction (Yuan)
1	0	0	754,759	8,384	16768
2	0	0	748,608	14,536	29,071
3	1	20,000	716,367	26,776	73,553
4	2	40,000	711,319	11,824	63,648

A comprehensive analysis of Table 4 and Figure 7 reveals that in the short term, the number of additional standby dryers exerts a direct and substantial influence on post-training cost reduction

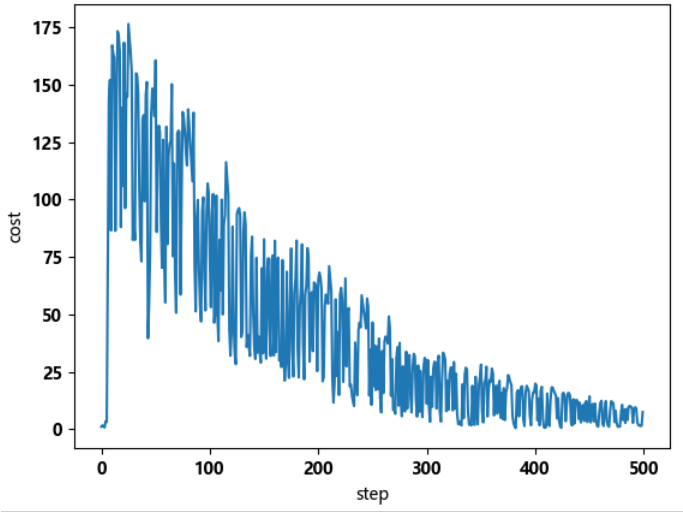

Figure 5. Loss function curve.

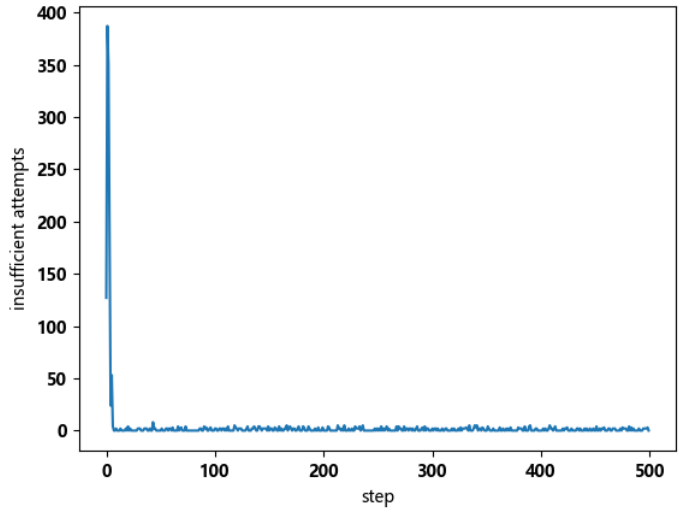

Figure 6. Insufficient drying unit count.

outcomes. However, as the cycle duration extends, the advantages conferred by supplementary standby dryers gradually become more pronounced, with a higher number of such units accelerating the rate of electricity cost decline. For instance, when comparing the third and fourth combinations illustrated in Figure 7, the third combination outperforms the fourth for total operational durations of up to two years; beyond a cumulative operating period of two and a half years, however, the fourth combination becomes the more advantageous option.

Moreover, an increase in the number of standby dryers reduces the complexity of deriving an optimal scheduling strategy. Configurations with an insufficient number of standby dryers necessitate more training iterations and extended training periods, and do not consistently yield satisfactory scheduling outcomes. By augmenting the number of dryers, the action space expands, thereby increasing the range of action combinations available to the intelligent agent and enhancing the efficiency of the scheduling strategy's resolution.

Figure 8 and Table 5 represent two distinct formats of the scheduling output. Figure 8 provides a

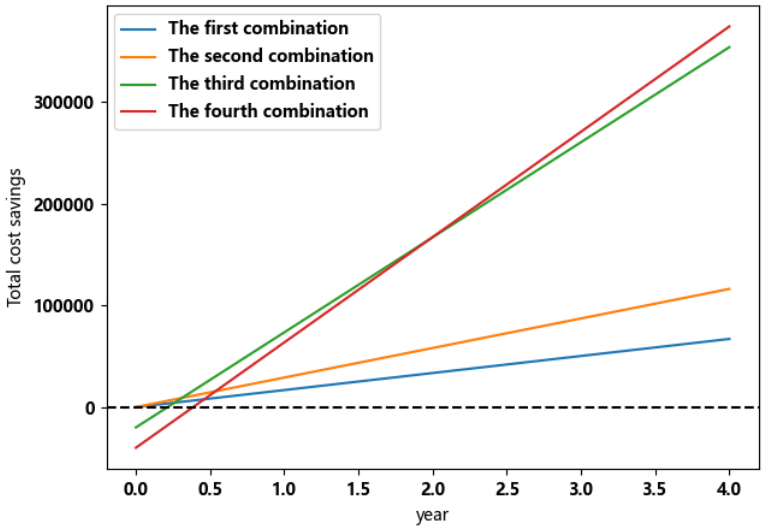

Figure 7. Summary of cost reductions across various combinations.

macroscopic overview of the internal scheduling process within the dryer cluster, while Table 5 offers a detailed account, precisely delineating the state of the two tanks within each dryer at any given time node.

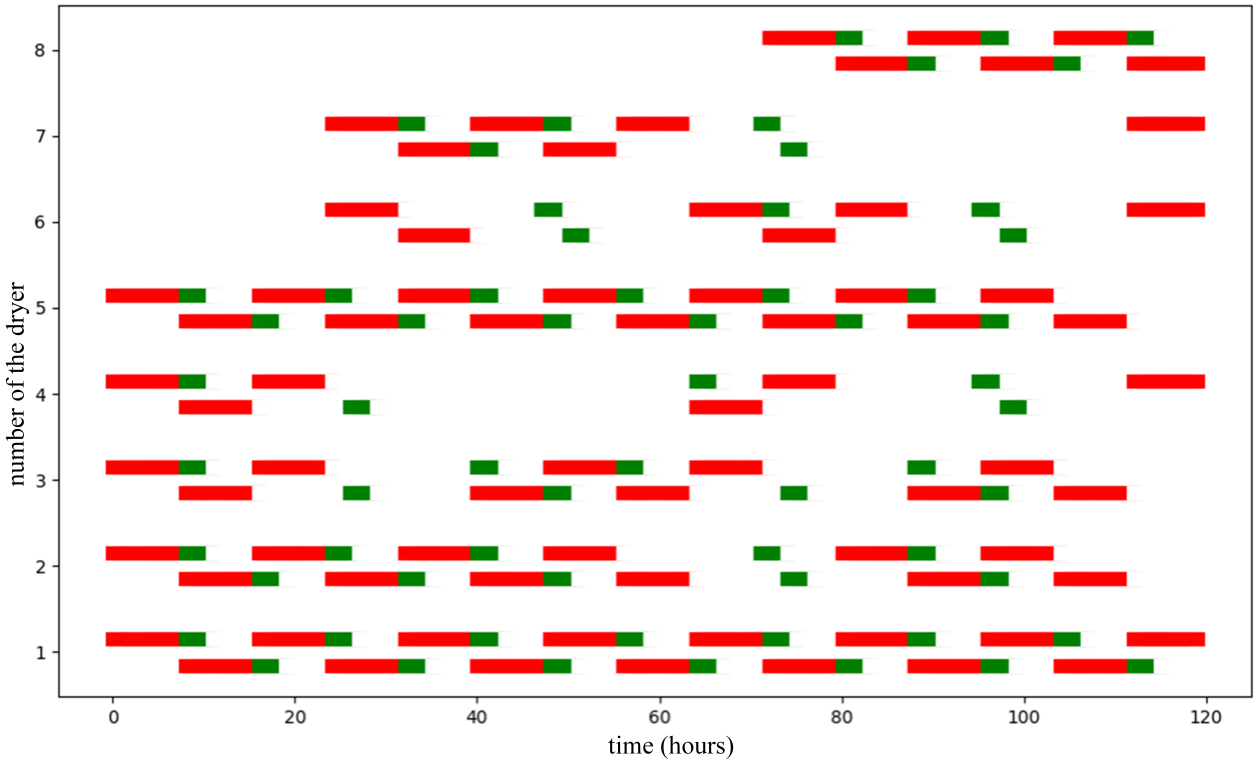

Figure 8. Drying unit ensemble scheduling Gantt chart.

Figure 8 displays the Gantt chart for the scheduling of eight dryers over the first 120 hours following training. The horizontal axis represents time nodes, while the vertical axis indicates the dryer identification numbers. Each dryer is depicted with two horizontal lines corresponding to the two tanks within the dryer. Red segments denote that the respective tank is engaged in gas drying operations, with a length of 8 representing 8 hours of operation, whereas green segments indicate that the tank is undergoing regeneration, with a length of 3 representing 3 hours of the regeneration process.

Table 5. Drying unit internal state table.

state	adsorber1	adsorber2
0	work	0
1	work	0
2	work	0
3	work	0
4	work	0
5	work	regen
6	work	regen
7	work	regen

Table 5 presents the internal state table for Dryer 1 after training; the program outputs eight analogous tables, one for each dryer. The two columns, "adsorber1" and "adsorber2," correspond to the two tanks within the dryer, while each row represents a discrete time node. The entries in the table comprise three states—"regen," "work," and "0"—indicating that the corresponding tank is engaged in heating regeneration, performing gas drying operations, or is in a delay/no-action state, respectively. For example, during time nodes 8 to 10, the table indicates that Dryer 1's adsorber1 is undergoing heating regeneration, while adsorber2 is actively engaged in gas drying operations.

5.3. Comparative and ablation studies

To comprehensively evaluate the efficacy of our proposed deep reinforcement learning (DRL) framework, we conducted a series of comparative and ablation experiments. These studies aim to (1) benchmark our method against established baseline algorithms and (2) isolate and verify the contribution of key components within our model architecture.

(1) Performance Comparison with Baseline Methods

We compared our DRL-based approach against two representative baseline methods to contextualize its performance, the rule-based greedy (RBG) strategy and genetic algorithm (GA). The RBG strategy is a heuristic algorithm that embodies a simple, intuitive policy. This strategy mandates that all heating regeneration operations for the dryers must be scheduled exclusively during the off-peak electricity price periods. If no off-peak slots are available when a dryer requires regeneration, the dryer remains idle until the next off-peak window, potentially violating production requirements. The GA is a well-established metaheuristic optimization algorithm widely used for scheduling problems. We implemented a customized GA where a chromosome represents a complete scheduling plan for all dryers over the 180-day cycle. The fitness function was designed to minimize the total electricity cost, with a significant penalty term added for any time step where the number of active dryers fell below the operational requirement. We tuned the GA's parameters (population size = 100, crossover rate =

0.8, mutation rate = 0.1) through preliminary experiments to ensure its best possible performance.

The comparison was conducted under the "1 additional standby dryer" scenario (Combination 3 from Table 4), which our initial analysis identified as the most cost-effective. The results, summarized in Table 6, evaluate each method based on total electricity cost and task completion reliability. The results clearly indicate the superiority of our proposed DRL approach. The RBG strategy, while conceptually simple, proved highly unreliable, failing to meet production demands for nearly a quarter of the operational time. The GA performed significantly better, reducing costs and improving reliability, but it still struggled to eliminate task failures completely, likely due to becoming trapped in local optima. In contrast, our DRL method not only achieved the lowest electricity cost, saving an additional 3% compared to the GA, but also demonstrated exceptional reliability with a near-zero task failure rate. This highlights the DRL agent's ability to learn a complex, dynamic policy that effectively navigates the trade-offs between cost and operational constraints.

Table 6. Performance comparison with baseline methods (180-day cycle).

Method	Avg. Total Electricity Cost (Yuan)	Task Failure Rate (%)
Rule-Based Greedy (RBG)	795,210	24.5%
Genetic Algorithm (GA)	738,450	2.8%
Proposed DRL	716,367	0.12%

(2) Ablation Study

To validate the architectural design of our multi-agent system, we conducted an ablation study to investigate the specific contribution of the global reward function. This component was designed to foster cooperative behavior among the otherwise independent dryer agents, ensuring the collective task of meeting production demand is fulfilled.

For this study, we created a variant of our model, termed "DRL without Global Reward" in which the global reward component was deactivated. In this configuration, each agent's learning process is driven solely by its individual reward, which is based only on the electricity tariff at the time of its action. We trained this ablated model under the same conditions as our full model and compared their performance.

The results in Table 6 unequivocally demonstrate that the global reward function is indispensable. The ablated model, driven by selfish incentives, achieved a marginally lower electricity cost. However, this was accomplished by systematically ignoring the system-level production requirements, leading to a task failure rate of 18.75%, which would be catastrophic in a real production environment. The agents learned to almost exclusively perform regeneration during off-peak hours, regardless of the consequences for the overall operation. Conversely, our full model, guided by the powerful negative feedback from the global reward when tasks failed, learned to establish a sophisticated cooperative policy. The agents successfully balanced their individual desire for cost savings with the critical need to maintain system-wide operational integrity. This confirms that the global reward signal is the crucial mechanism that transforms a group of independent, self-interested agents into an effective, cooperative team capable of solving the complex, constrained scheduling problem.

Table 7. Ablation study of the global reward function.

Model Configuration	Avg. Total Electricity Cost (Yuan)	Task Failure Rate (%)
Proposed DRL with Global Reward	716,367	0.12%
DRL without Global Reward	698,540	18.75%

6. Operational risk assessment and preventive control measures

A critical attribute of an effective industrial scheduling algorithm is its robustness—the ability to maintain performance and adapt in the face of unforeseen events and dynamic changes in operational requirements. To rigorously evaluate the reliability and flexibility of our proposed DRL-based approach, we move beyond static analysis and introduce a set of simulated dynamic scenarios. Rather than merely assessing risk, this section serves as a direct test of the algorithm's performance under fluctuating conditions, which are common in real-world factory environments. We establish an anticipated incident set and incorporate these events into the normal operational cycle. By examining the overall scheduling chart of the dryer cluster after training, we can directly observe the impact of these unforeseen incidents and assess the algorithm's capacity for real-time, adaptive control. The baseline configuration for this analysis is as follows: the factory operates 7 dryers, with 4 dryers required for standard operating conditions and 3 additional units on standby; the maximum gas drying operation time T_1 is 8 hours, and the minimum heating regeneration time T_2 is 3 hours.

Incident 1: During the development of the scheduling optimization strategy for the dryer cluster, external influences from factory-related projects necessitated adjustments in the number of dryers required by the operating conditions, all while maintaining the existing configuration of the dryer cluster. Consequently, at a certain point during training, the required number of dryers was altered.

Incident 2: During the normal process of solving the scheduling optimization strategy for the dryer cluster, certain dryers experienced malfunctions that prevented them from continuing gas drying operations. As a result, these faulty units were removed from the existing configuration without the addition of new dryers, leading to a change in the total number of dryers at a specific point during training.

After simulating the anticipated scenario of Incident 1, an increase of one required dryer was observed at the 60th time node (indicated by the blue marker). As shown in Figure 9, prior to this time node, there were consistently four red horizontal segments, signifying that four dryers were operating concurrently. Following the simulated incident, Dryer 7 was activated, resulting in five horizontal segments being present at every subsequent time node. This demonstrates that the program effectively addresses the issue introduced by Incident 1.

After simulating the anticipated scenario for Incident 2, Dryer No. 7 experienced a malfunction at a certain moment after the 60th time node (marked in blue) and was unable to continue gas drying operations. As shown in Figure 10, prior to the 60th time node, four red horizontal segments consistently indicated that four dryers were operating concurrently, with Dryer No. 7 functioning normally. Following the simulated incident, Dryer No. 7 ceased operations, and Dryer No. 5 was immediately activated, ensuring that four dryers continued to operate simultaneously at all times. This demonstrates that the program effectively addresses the issues introduced by Incident 2.

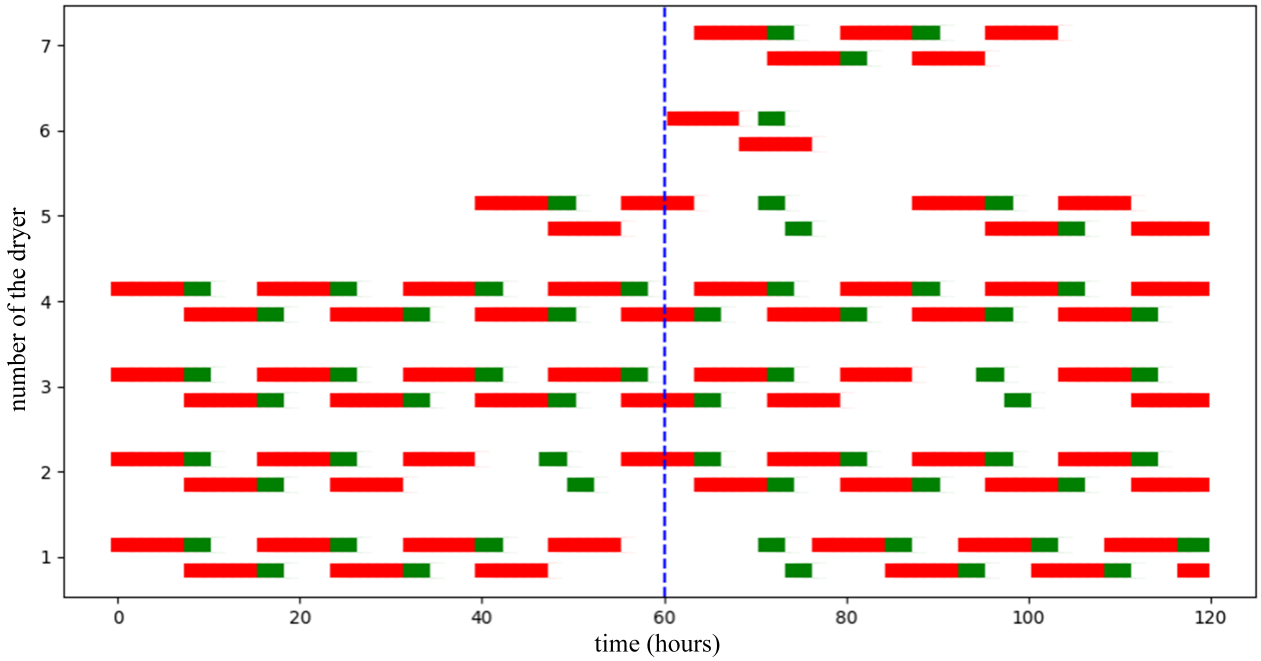

Figure 9. Gantt chart for the scheduling of simulated Incident 1.

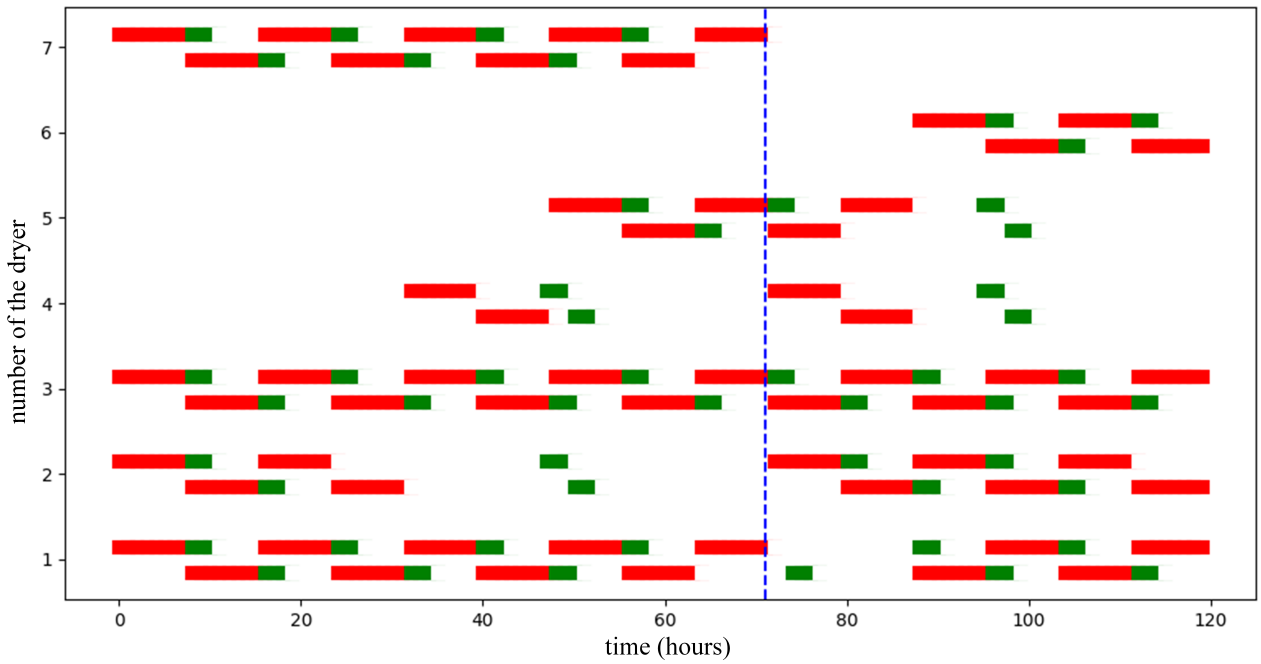

Figure 10. Gantt chart for the scheduling of simulated Incident 2.

7. Conclusions

This paper presents a dryer cluster scheduling optimization strategy algorithm based on the DQN approach. The scheduling problem is reformulated as a multi-agent cooperative challenge among multiple dryers. By employing the double-DQN algorithm, the method reduces the electricity consumption of each dryer and optimizes individual scheduling strategies through an integrated overall reward function, thereby achieving a reduction in total electricity costs without compromising the factory's overall gas drying operations. Moreover, the algorithm can recommend the number of additional dryers required based on the existing cluster configuration and project demands. Compared to traditional fixed algorithms used in the factory, the proposed method is more economical, flexible, and reliable; it not only meets the goal of lowering electricity expenses but also offers strategic guidance for production expansion while mitigating budget recovery risks. Furthermore, the paper introduces two anticipated incident scenarios related to the dryer cluster scheduling problem and conducts simulation experiments to assess the algorithm's resilience. Through computational analysis, operational risk assessment, and incident handling, the method has been validated as effective, superior, and reliable. By addressing the dryer cluster scheduling optimization, this study offers a novel research perspective and a viable solution approach for similar combinatorial optimization problems in unit cluster configurations.

Furthermore, this study provides a foundational framework that can be extended to address more complex industrial scenarios. A key avenue for future research is the integration of peak demand management into the optimization model. While the grid capacity cost was justifiably excluded in this specific case due to the dryers' stable power profile, applying this multi-agent DRL approach to equipment with higher and more volatile power consumption would necessitate its inclusion. Future work could enhance the objective function to incorporate penalties for high peak loads, thereby training the agents to not only leverage TOU tariffs but also to cooperatively avoid simultaneous operation of high-power processes. This would yield a more holistic energy cost optimization strategy applicable to a broader range of industrial facilities.

Author contributions

Guoliang Feng: Supervision, Methodology, Writing; Tianren Gao: Conceptualization, Methodology, Writing; Shaojun Bian: Supervision, Review; Tianming Yu: Project administration, Editing.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Science and Technology Project of Jilin Province under Grant 20240101354JC.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Dauzère-Pérès S, Ding J, Shen L, Tamssaouet K (2024) The flexible job shop scheduling problem: A review. *Eur J Oper Res* 314: 409–432. https://doi.org/10.1016/j.ejor.2023.05.017
2. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. *Multimed Tools Appl* 80: 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
3. Gad AG (2022) Particle swarm optimization algorithm and its applications: a systematic review. *Arch Comput Method Eng* 29: 2531–2561. https://doi.org/10.1007/s11831-021-09694-4
4. Nayar N, Gautam S, Singh P, Mehta G (2021) Ant colony optimization: A review of literature and application in feature selection. *Inventive Computation and Information Technologies: Proceedings of ICICIT 2020*, 285–297. https://doi.org/10.1007/978-981-33-4305-4_22
5. Wu M, Yang D, Liu T (2022) An improved particle swarm algorithm with the elite retain strategy for solving flexible jobshop scheduling problem. In *Journal of Physics: Conference Series* 2173: 012082. IOP Publishing. https://doi.org/10.1088/1742-6596/2173/1/012082
6. Xu X, Wang L (2021) An improved gaming particle swarm algorithm based the rules of flexible job shop scheduling. In *2021 7th International Conference on Systems and Informatics (ICSAI)*, 1–5. IEEE. https://doi.org/10.1109/ICSAI53574.2021.9664124
7. Tutumlu B, Saraç T (2023) A mip model and a hybrid genetic algorithm for flexible job-shop scheduling problem with job-splitting. *Comput Oper Res* 155: 106222. https://doi.org/10.1016/j.cor.2023.106222
8. Zhang G, Hu Y, Sun J, Zhang W (2020) An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. *Swarm Evol Comput* 54: 100664. https://doi.org/10.1016/j.swevo.2020.100664
9. Liu Z, Wang J, Zhang C, Chu H, Ding G, Zhang L (2021) A hybrid genetic-particle swarm algorithm based on multilevel neighbourhood structure for flexible job shop scheduling problem. *Comput Oper Res* 135: 105431. https://doi.org/10.1016/j.cor.2021.105431
10. Fontes DBMM, Homayouni SM, Gonçalves JF (2023) A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources. *Eur J Oper Res* 306: 1140–1157. https://doi.org/10.1016/j.ejor.2022.09.006
11. Amirteimoori A, Mahdavi I, Solimanpur M, Ali SS, Tirkolaee EB (2022) A parallel hybrid pso-ga algorithm for the flexible flow-shop scheduling with transportation. *Comput Ind Eng* 173: 108672. https://doi.org/10.1016/j.cie.2022.108672
12. Torres-Tapia W, Montoya-Torres JR, Ruiz-Meza J, Belmokhtar-Berraf S (2022) A matheuristic based on ant colony system for the combined flexible jobshop scheduling and vehicle routing problem. *IFAC-PapersOnLine* 55: 1613–1618. https://doi.org/10.1016/j.ifacol.2022.09.621

13. Vivek S, Rakesh K, Mohan BR (2022) Optimized distributed job shop scheduling using balanced job allocation and modified ant colony optimization. In *Pattern Recognition and Data Analysis with Applications*, 271–281. Springer. https://doi.org/10.1007/978-981-19-1520-8_21

14. Fan J, Zhang C, Liu Q, Shen W, Gao L (2022) An improved genetic algorithm for flexible job shop scheduling problem considering reconfigurable machine tools with limited auxiliary modules. *J Manuf Syst* 62: 650–667. https://doi.org/10.1016/j.jmsy.2022.01.014

15. Sun K, Zheng D, Song H, Cheng Z, Lang X, Yuan W, Wang J (2023) Hybrid genetic algorithm with variable neighborhood search for flexible job shop scheduling problem in a machining system. *Expert Syst Appl* 215: 119359. https://doi.org/10.1016/j.eswa.2022.119359

16. Vela A, Cruz-Duarte JM, carlos Ortiz-Bayliss J, Amaya I (2021) Tailoring job shop scheduling problem instances through unified particle swarm optimization. *IEEE Access* 9: 66891–66914. https://doi.org/10.1109/ACCESS.2021.3076426

17. Anuar NI, Fauadi MHFM (2021) A study on multi-objective particle swarm optimization in solving job-shop scheduling problems. *International Journal of Computer Information Systems and Industrial Management Applications* 13: 11–11.

18. Lachtar N, Driss I (2023) Application of ant colony optimization for job shop scheduling in the pharmaceutical industry. *Journal Européen des Systèmes Automatisés* 56. https://doi.org/10.18280/jesa.560501

19. Matrenin PV (2022) Improvement of ant colony algorithm performance for the job-shop scheduling problem using evolutionary adaptation and software realization heuristics. *Algorithms* 16: 15. https://doi.org/10.3390/a16010015

20. Alpaydin E (2021) *Machine learning*, MIT press.

21. Gupta V, Mishra VK, Singhal P, Kumar A (2022) An overview of supervised machine learning algorithm. In *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 87–92. IEEE. https://doi.org/10.1109/SMART55829.2022.10047618

22. Naeem S, Ali A, Anam S, Ahmed MM (2023) An unsupervised machine learning algorithms: Comprehensive review. *International Journal of Computing and Digital Systems*.

23. Yang X, Song Z, King I, Xu Z (2022) A survey on deep semi-supervised learning. *IEEE T Knowl Data En* 35: 8934–8954. https://doi.org/10.1109/TKDE.2022.3220219

24. Abel D, Barreto A, Van Roy B, Precup D, van Hasselt HP, Singh S (2023) A definition of continual reinforcement learning. *Advances in Neural Information Processing Systems* 36: 50377–50407.

25. Sharifani K, Amini M (2023) Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal* 10: 3897–3904.

26. Li Y (2022) Research and application of deep learning in image recognition. In *2022 IEEE 2nd international conference on power, electronics and computer applications (ICPECA)*, 994–999. IEEE. https://doi.org/10.1109/ICPECA53709.2022.9718847

27. Huaysrijan A, Pongpinigpinyo S (2021) Deep convolution neural network for thai classical music instruments sound recognition. In *2021 25th International Computer Science and Engineering Conference (ICSEC)*, 283–288. IEEE. https://doi.org/10.1109/ICSEC53205.2021.9684611

28. Alsobhani A, ALabboodi HMA, Mahdi H (2021) Speech recognition using convolution deep neural networks. In *Journal of Physics: Conference Series* 1973: 012166. IOP Publishing. https://doi.org/10.1088/1742-6596/1973/1/012166

29. Tercan H, Meisen T (2022) Machine learning and deep learning based predictive quality in manufacturing: a systematic review. *J Intell Manuf* 33: 1879–1905. https://doi.org/10.1007/s10845-022-01963-8

30. Zamora-Hernandez MA, Castro-Vargas JA, Azorin-Lopez J, Garcia-Rodriguez J (2021) Deep learning-based visual control assistant for assembly in industry 4.0. *Comput Ind* 131: 103485. https://doi.org/10.1016/j.compind.2021.103485

31. Subramanian N, Elharrouss O, Al-Maadeed S, Chowdhury M (2022) A review of deep learning-based detection methods for covid-19. *Comput Biol Med* 143: 105233. https://doi.org/10.1016/j.compbiomed.2022.105233

32. Kapoor S, Narayanan A (2023) Leakage and the reproducibility crisis in machine-learning-based science. *Patterns* 4. https://doi.org/10.1016/j.patter.2023.100804

33. Ladosz P, Weng L, Kim M, Oh H (2022) Exploration in deep reinforcement learning: A survey. *Inform Fusion* 85: 1–22. https://doi.org/10.1016/j.inffus.2022.03.003

34. Lv L, Zhang C, Fan J, Shen W (2025) Deep reinforcement learning for job shop scheduling problems: A comprehensive literature review. *Knowledge-Based Syst*, 113633. https://doi.org/10.1016/j.knosys.2025.113633

35. Wang J, Li Y, Zhang Z, Wu Z, Wu L, Jia S, Peng T (2024) Dynamic integrated scheduling of production equipment and automated guided vehicles in a flexible job shop based on deep reinforcement learning. *Processes* 12: 2423. https://doi.org/10.3390/pr12112423

36. Zulfiqar M, ul Abdeen Z, Kamran M (2024) Optimizing electric vehicle charging scheduling using enhanced multi-agent neural networks with dynamic pricing. *J Energy Storage* 99: 113317. https://doi.org/10.1016/j.est.2024.113317

AIMS Press